

### **IN THE CLAIMS**

Please amend the claims as follows:

1. (Currently Amended) A method to be performed by a data processing system to improve fault tolerance comprising:

providing distributed queuing of workflows, whose execution is requested by one or more execution-requesting clients, among a plurality of workflow engines; [[and]]

only if a workflow is successfully completed by a first workflow engine for an execution-requesting client, sending a notification to the execution-requesting client, else assigning the workflow to a subsequent ~~second~~ workflow engine by sending it a work assignment message, in response to which the subsequent ~~second~~ workflow engine alone performs ~~completes~~ the workflow; and

sending a notification to the execution-requesting client only if the workflow is successfully completed by the subsequent workflow engine.

2. (Original) The method recited in claim 1, wherein providing is performed by a load manager.

3. (Original) The method recited in claim 2, wherein the load manager comprises a commercially available middleware product.

4. (Previously Presented) The method recited in claim 1, wherein the notification is performed by a certified messaging capability.

5. (Original) The method recited in claim 4, wherein the certified messaging capability is performed by a load manager.

- 
6. (Original) The method recited in claim 4, wherein the load manager comprises a commercially available middleware product.
7. (Original) The method recited in claim 4, wherein the certified messaging capability is performed by a certified message receiver forming part of the workflow.
8. (Currently Amended) The method recited in claim 4 and further comprising:  
the certified messaging capability sending the ~~the~~ [[a]] notification to the execution-requesting client only if the workflow is successfully completed by the subsequent ~~second~~ workflow engine.
9. (Currently Amended) A method to be performed by a computer network comprising a plurality of clients and a plurality of workflow engines:  
providing distributed queuing of workflows, whose execution can be requested by one or more execution-requesting clients, among the plurality of workflow engines; and  
determining whether a workflow has been successfully completed by a first workflow engine on behalf of an execution-requesting client; ~~[[and]]~~  
only if so, sending a notification to the execution-requesting client;  
otherwise, assigning the workflow to a subsequent ~~second~~ workflow engine by sending it a work assignment message, and the subsequent ~~second~~ workflow engine alone performing ~~completing~~ the workflow; and  
sending a notification to the execution-requesting client only if the workflow is successfully completed by the subsequent workflow engine.
10. (Original) The method recited in claim 9, wherein providing is performed by a load manager.
11. (Original) The method recited in claim 10, wherein the load manager comprises a commercially available middleware product.

12. (Original) The method recited in claim 9, wherein sending is performed by a certified messaging capability.
13. (Original) The method recited in claim 12, wherein the certified messaging capability is performed by a load manager.
14. (Original) The method recited in claim 12, wherein the load manager comprises a commercially available middleware product.
15. (Original) The method recited in claim 12, wherein the certified messaging capability is performed by a certified message receiver in the workflow.
16. (Currently Amended) The method recited in claim 12 and further comprising:  
the certified messaging capability sending the [[a]] notification to the execution-requesting client only if the workflow is successfully completed by the subsequent ~~second~~ workflow engine.

17. (Currently Amended) A computer adapted for use in a computer network comprising a plurality of workflow engines, the computer executing a computer program, the computer program operating the computer in a fault-tolerant manner and comprising the operations of:

requesting a workflow execution on behalf of a client;

a distributed queuing capability assigning the workflow execution to a first workflow engine;

determining whether the workflow execution has been successfully completed by the first workflow engine; [[and]]

only if so, sending a notification to the client;

otherwise, assigning the workflow execution to a subsequent ~~second~~ workflow engine by sending it a work assignment message, and the subsequent ~~second~~ workflow engine alone performing ~~completing~~ the workflow; and

sending a notification to the client only if the workflow is successfully completed by the subsequent workflow engine.

18. (Original) The computer recited in claim 17, wherein requesting is performed by a load manager.

19. (Original) The computer recited in claim 17, wherein sending is performed by a certified messaging capability.

20. (Original) The computer recited in claim 19, wherein the certified messaging capability is performed by a certified message receiver in the first workflow engine.

21. (Currently Amended) The computer recited in claim 19 and further comprising:  
the certified messaging capability sending the ~~the~~ notification to the client only if the workflow execution is successfully completed by the subsequent ~~second~~ workflow engine.

22. (Currently Amended) A computer network comprising:
- a plurality of clients;
  - a plurality of workflow engines; and
  - at least one computer program, the computer program operating in a fault-tolerant manner and performing the operations of:
    - requesting a workflow execution on behalf of a client;
    - assigning the workflow execution to a first workflow engine;
    - determining whether the workflow execution has been successfully completed by the first workflow engine; [[and]]
    - only if so, sending a notification to the client;
    - otherwise, assigning the workflow execution to a subsequent ~~second~~ workflow engine by sending it a work assignment message, and the subsequent ~~second~~ workflow engine alone performing completing the workflow; and.
    - sending a notification to the client only if the workflow is successfully completed by the subsequent workflow engine.
23. (Previously Presented) The computer network recited in claim 22, wherein requesting is performed by a load manager having a distributed queuing capability.
24. (Original) The computer network recited in claim 22, wherein sending is performed by a certified messaging capability.
25. (Original) The computer network recited in claim 24, wherein the certified messaging capability is performed by a certified message receiver in the first workflow engine.
26. (Currently Amended) The computer network recited in claim 24 and further comprising:
- the certified messaging capability sending the [[a]] notification to the client only if the workflow execution is successfully completed by the subsequent ~~second~~ workflow engine.

27. (Currently Amended) A computer-readable medium containing computer instructions for instructing a processor, the processor adapted for use in a computer network comprising a plurality of workflow engines, wherein the instructions comprise:

requesting a workflow execution on behalf of a client;

a distributed queuing capability assigning the workflow execution to a first workflow engine;

determining whether the workflow execution has been successfully completed by the first workflow engine; [[and]]

only if so, sending a notification to the client;

otherwise, assigning the workflow execution to a subsequent ~~second~~ workflow engine by sending it a work assignment message, and the subsequent ~~second~~ workflow engine alone completing the workflow; and

sending a notification to the client only if the workflow is successfully completed by the subsequent workflow engine.

28. (Original) The computer-readable medium recited in claim 27, wherein requesting is performed by a load manager.

29. (Original) The computer-readable medium recited in claim 27, wherein sending is performed by a certified messaging capability.

30. (Original) The computer-readable medium recited in claim 29, wherein the certified messaging capability is performed by a certified message receiver in the first workflow engine.

31. (Currently Amended) The computer-readable medium recited in claim 29 and further comprising:

the certified messaging capability sending the [[a]] notification to the client only if the workflow execution is successfully completed by the subsequent ~~second~~ workflow engine.

32. (Currently Amended) An article comprising a machine-accessible medium having instructions for instructing a processor forming part of a plurality of workflow engines, wherein the instructions, when accessed, result in a machine performing:

requesting a workflow execution on behalf of a client;

assigning the workflow execution to a first workflow engine;

determining whether the workflow execution has been successfully completed by the first workflow engine; [[and]]

only if so, sending a notification to the client;

otherwise, assigning the workflow execution to a subsequent ~~second~~ workflow engine by sending it a work assignment message, and the subsequent ~~second~~ workflow engine alone completing the workflow; and

sending a notification to the client only if the workflow is successfully completed by the subsequent workflow engine.

33. (Previously Presented) The article recited in claim 32, wherein requesting is performed by a load manager having a distributed queuing capability.

34. (Previously Presented) The article recited in claim 32, wherein sending is performed by a certified messaging capability.

35. (Previously Presented) The article recited in claim 34, wherein the certified messaging capability is performed by a certified message receiver in the first workflow engine.

36. (Currently Amended) The article recited in claim 34 and further comprising:  
the certified messaging capability sending the ~~the~~ notification to the client only if the workflow execution is successfully completed by the subsequent ~~second~~ workflow engine.